Engineering Data Analysis with Matlab

Anke Scherb

Engineering Risk Analysis Group
Technische Universität München

Phone: 089 289 23013
E-mail: anke.scherb@tum.de

Engineering Data Analysis with Matlab

Today's lecture

- Matlab functions
- Random variables
- Distribution fitting

Matlab – Functions

- M-files that provide output as a function of input (e.g. data variables)
- Called by other Matlab programs or from command window
- Types:
  - Built-in functions, e.g. sin, cos, log, …
  - User-defined functions

Matlab – Functions

- The definition of new functions must be stored in a file with name *function_name.m*

- Function syntax, first executable line of the function:

```
function [output1 output2 …] = function_name(input1, input2, …)

Matlab statements
```

- Declare a function named by "function_name" that accepts inputs (input1, input2,…) and returns outputs (output1, output2, …)

Matlab – Functions

Example: Function *fun* stored in file *fun.m*

```
function y = fun(x)

y = x^2 - 1;
```

Call function in main program

```
…

b = 1.5;

f_b = fun(b);

…
```

Example: fun.m

Matlab – Functions

Function with two outputs - Example: Function *stat* stored in file *stat.m*

```matlab
function [mean, stdev] = stat(data)

mean = sum(data)/length(data);

var = sum((data-mean).^2)/(length(data) - 1);

stdev = sqrt(var);
```
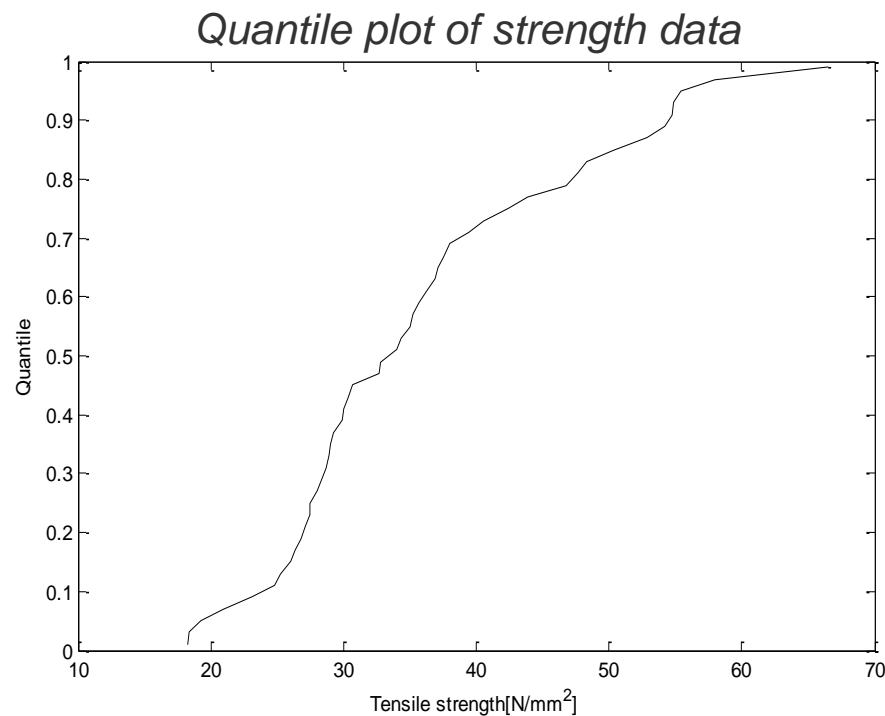
Note:

- All variables within the body of a function are local variables
- A function does not necessarily need to provide an output, e.g. plotting functions

Example: stat.m

Quantile plot – Plots the values below which a certain fraction of the samples fall

- Sort samples in ascending order $x_1, x_2, \ldots, x_n$
- Plot the pairs x_i and $(i - 0.5)/n$

Quantile plot of strength data

Example: qtlplot.m

7

Matlab – Function handles

- Callable association to a Matlab function; a MATLAB data type that stores an association to a function.

```
function_handle = @function_name
```

Function handles can be used for passing a function to another function

```
h = @sin;

h(pi)
```

Matlab – Anonymous Functions

- Simple functions that are not saved in separate files but is associated with a variable whose data type is function_handle

```
function_handle = @(input1, input2, …) expression
```

Example:

```
fun = @(x) x^2 – 1;

fzero(fun,0.5)
```

Note: Matlab build-in functions:

- **fzero (fun, x0)** finds a root of fun near x0

Example: fzero.m

Matlab – Structures

- Collection of data of different types that represent a single idea or "object"

Example:

```
data.name = 'strength';

data.values = [37.1 36.9 40.5 46.8 37.6];

data.mean = mean(data.values);

data.std = std(data.values);
```

- Note: Structures can also be created using the `struct()` function

Example: struct_.m

Random variables

Mathematical tool for modeling uncertain quantities

- Random variables map possible outcomes of an experiment to the real numbers
- A random variable is represented by a capital letter, e.g. X
- An outcome of a random variable is represented by a lower case letter, e.g. x
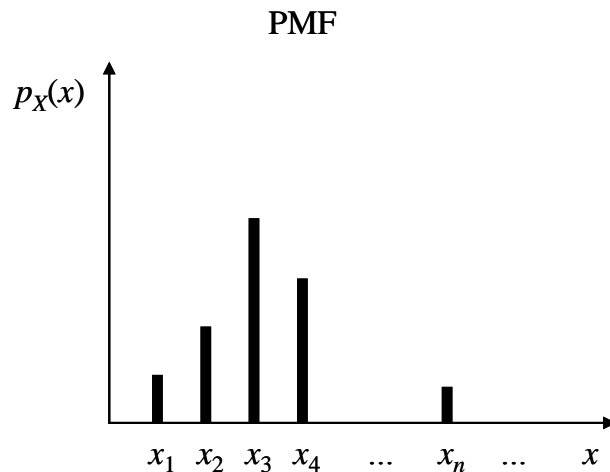
Discrete random variables

Random variables with discrete number of outcomes

Probability Mass Function (PMF) – Function that describes the probability that a discrete random variable X takes a specific value x

$$p_X(x) = \Pr(X = x)$$

CDF of discrete random variable: $\quad F_X(x) = \sum_{x_i \leq x} p_x(x_i)$

PMF

$p_X(x)$

$x_1 \; x_2 \; x_3 \; x_4 \quad \ldots \quad x_n \quad \ldots \quad x$

CDF

$F_X(x)$

1.0

x
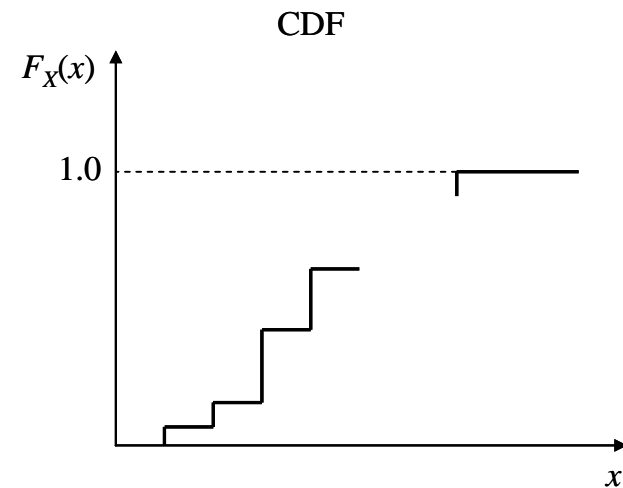
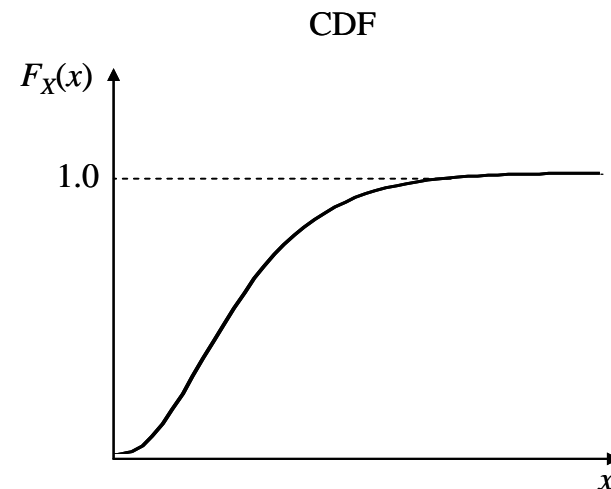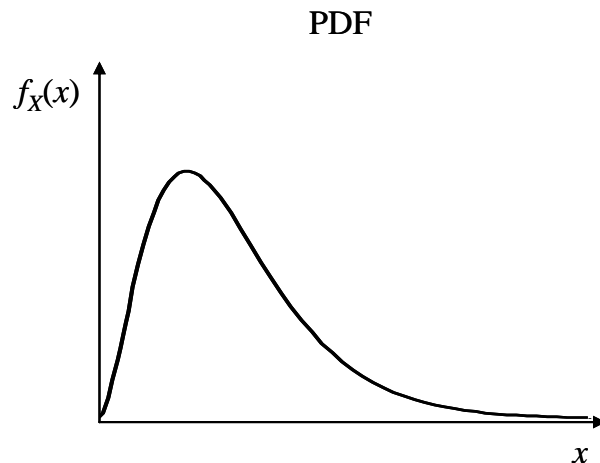## Continuous random variables

Random variables which can take any value within one or more intervals

Probability Density Function (PDF) – Function whose integral over an interval gives the probability that X takes a value within this interval

$$f_X(x)dx = \Pr(x < X \le x + dx)$$

CDF of continuous random variable: $F_X(x) = \int_{-\infty}^{x} f_x(t)dt$

PDF

$f_X(x)$

x

CDF

$F_X(x)$

1.0

x

Matlab – Random variables

Probability Density Function (PDF) – Function whose integral over an interval gives the probability that X takes a value within this interval

$$f_X(x)dx = \Pr(x < X \leq x + dx)$$

```
pdf('name',x,par1,par2,…)
```

```
namepdf(x,par1,par2,…)
```

Note: If `'name'` is a distribution that describes a discrete random variable then `pdf` returns the PMF

Matlab – Random variables

Cumulative distribution function (CDF) – Function that describes the
probability that a random variable X is smaller than or equal to an outcome x

$$F_X(x) = \Pr(X \leq x)$$

```
cdf('name',x,par1,par2,…)
```

```
namecdf(x,par1,par2,…)
```

Notes:

- The CDF is a non-decreasing function
- The CDF has limits $F_X(-\infty) = 0$, $F_X(\infty) = 1$

Description of random variables

Mean value

- Discrete random variable

$$\mu_X = \mathrm{E}[X] = \sum_{\text{all } x_i} x_i p_X(x_i)$$

- Continuous random variable

$$\mu_X = \mathrm{E}[X] = \int_{-\infty}^{\infty} x \cdot f_X(x) dx$$

Variance

$$\mathrm{Var}[X] = \mathrm{E}[(X - \mu_X)^2]$$

- Discrete random variable

$$\mathrm{Var}[X] = \sum_{\text{all } x_i} (x_i - \mu_X)^2 p_X(x_i)$$

- Continuous random variable

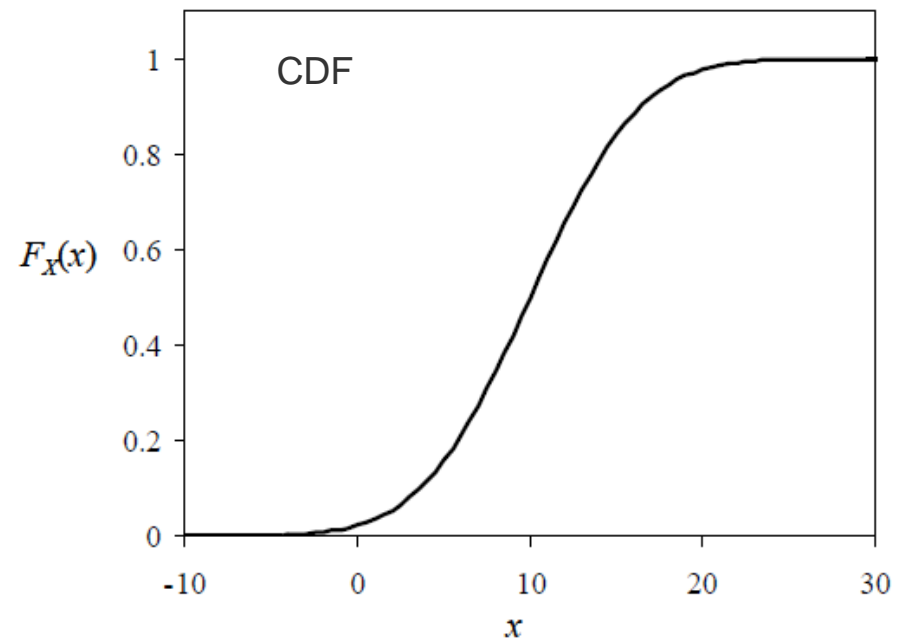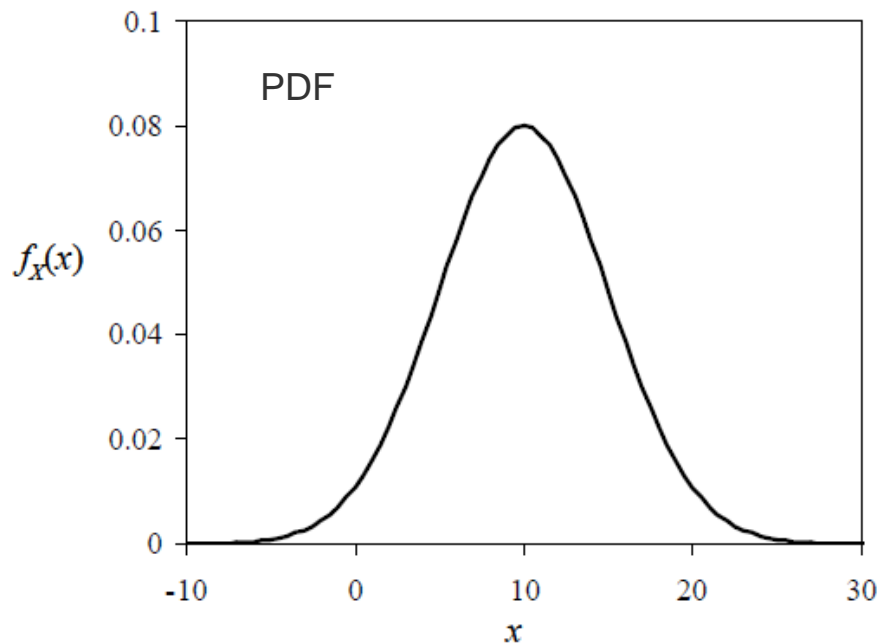$$\mathrm{Var}[X] = \int_{-\infty}^{\infty} (x - \mu_X)^2 f_X(x) dx$$

Standard deviation

$$\sigma_X = \sqrt{\mathrm{Var}[X]}$$

Distribution types - Examples

Normal (Gaussian) distribution

- Defined by two parameters, the mean and the standard deviation: $X \sim N(\mu, \sigma)$
- PDF is symmetric around the mean value

$$f_X(x) = \frac{1}{\sigma}\varphi\left(\frac{x-\mu}{\sigma}\right) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \qquad x \in (-\infty, +\infty)$$

$$\Phi(u) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}}\exp\left[-\frac{z^2}{2}\right]dz \qquad 17$$
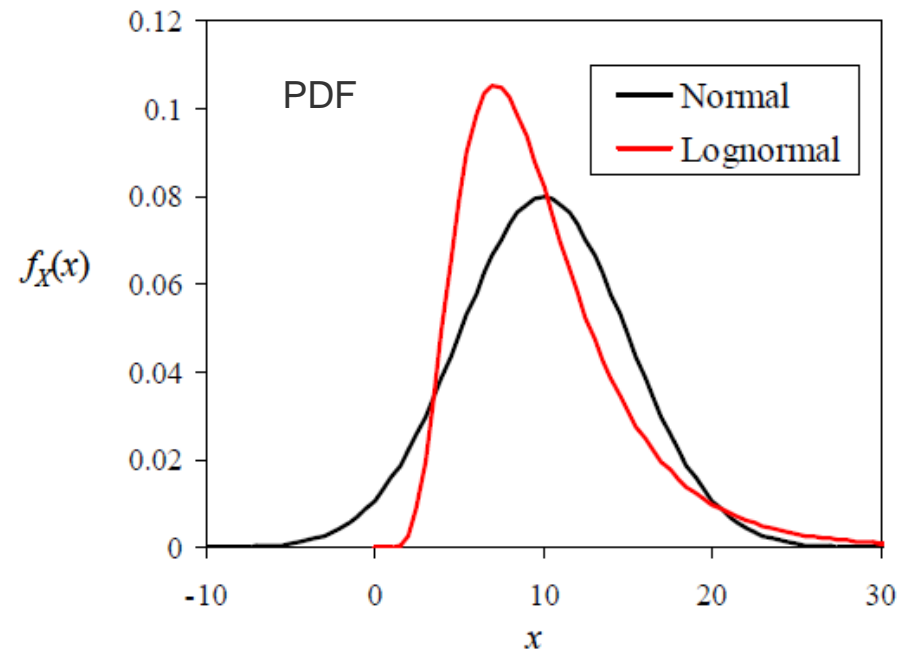
Distribution types - Examples

Lognormal distribution

- The logarithm of a lognormal random variable is normal
- Defined by two parameters
- Asymmetric distribution with strictly positive outcomes

$$f_X(x) = \frac{1}{\zeta x} \varphi\left(\frac{\ln x - \lambda}{\zeta}\right) \qquad x \in (0, +\infty)$$
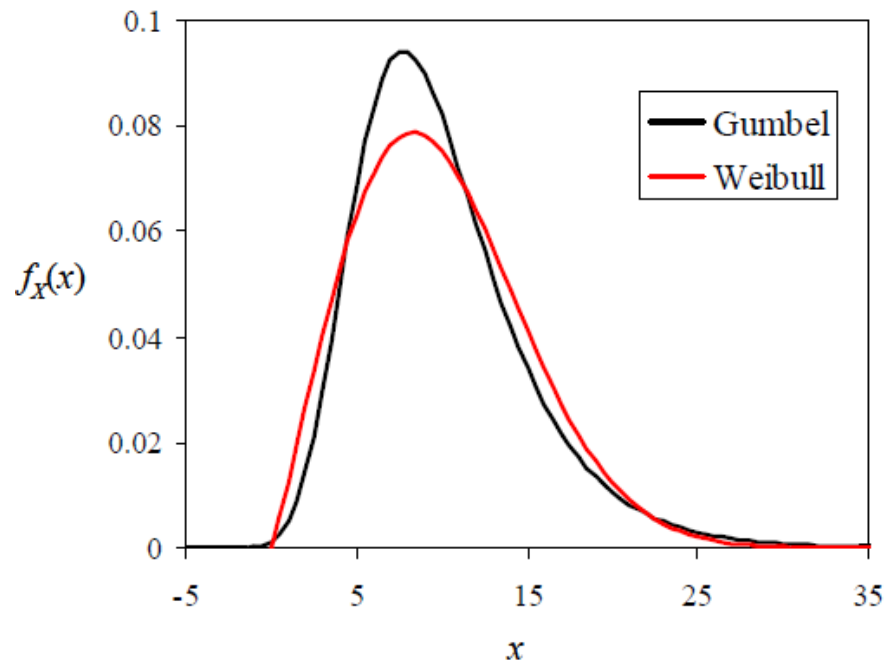
$$F_X(x) = \Phi\left(\frac{\ln x - \lambda}{\zeta}\right)$$

Distribution types - Examples

Extreme value distributions – Model maxima or minima of random variables

- Gumbel distribution: Models maxima of distributions with exponential tail
- Weibull distribution: Models minima of distributions with zero lower bound

Matlab - Distribution types

Distributions available in Matlab

`'norm' or 'normal'`	Normal distribution
`'logn' or 'lognormal'`	Lognormal distribution
`'beta' or 'Beta'`	Beta distribution
`'ev' or 'Extreme Value'`	Gumbel or Extreme value distribution
`'wbl' or 'Weibull'`	Weibull distribution
…	…

Matlab - Distribution types

Distribution statistics

```
namestat(par1,par2,…)
```

Computes the mean and variance of the random variable with distribution `name` and parameters `par1, par2,` …

Matlab - Distribution objects

- Construct a distribution object using `makedist`
- Collect parameters and data sets related to a particular distribution type
- The CDF/PDF and statistics can be evaluated using available methods (`cdf,pdf,mean,var,std,`…)

Example:

```
pn = makedist('norm','mu',10,'sigma',5);

cdf(pn,x)

pdf(pn,x)

mean(pn)

var(pn)
```

Example: distrib.m

Distribution fitting

- Find the suitable distribution and its parameters to model an uncertain quantity for which experimental data (observations) are available
- Aim: predict the probability or forecast the frequency of occurrence of the value of the phenomenon; find the distribution type that fits the data best.

Method of moments

- Compute the sample statistics (sample mean and sample standard deviation)

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(x_i - \bar{x}\right)^2}$$

- **Idea: Find the distribution parameters by requiring that the mean and standard deviation of the distribution are equal to** $\bar{x}$ **and** s

Distribution fitting

Method of moments – Example

- Fitting of Timber tensile strength data to the **Weibull** distribution

Weibull CDF: $\quad F_X(x) = 1 - \exp\left[-\left(\dfrac{x}{u} \right)^k \right], \qquad x \geq 0$

Mean: $\qquad\qquad \mu = u \cdot \Gamma\left(1 + \dfrac{1}{k} \right)$

Standard deviation: $\quad \sigma = u \cdot \sqrt{ \Gamma\left(1 + \dfrac{2}{k} \right) - \Gamma^2\left(1 + \dfrac{1}{k} \right) }$

$\rightarrow \dfrac{\sigma}{\mu} = \dfrac{ \sqrt{ \Gamma\left(1 + \dfrac{2}{k} \right) - \Gamma^2\left(1 + \dfrac{1}{k} \right) } }{ \Gamma\left(1 + \dfrac{1}{k} \right) }$

- Γ: Gamma function => Matlab built-in function ***gamma()***

Distribution fitting

Method of moments – Example

- Fitting of Timber tensile strength data to the **Weibull** distribution

$$\textit{Weibull CDF:} \quad F_X(x) = 1 - \exp\left[-\left(\frac{x}{u}\right)^k\right], \qquad x \geq 0$$

$k =$ … use Matlab built-in functions fzero and gamma to find parameter k

$$u = \frac{\mu}{\Gamma(1 + \frac{1}{k})}$$

Example: wblfit_.m

Distribution fitting

Method of moments – Example

- Fitting of Timber tensile strength data to the **Lognormal** distribution

Lognormal CDF:
$$F_X(x) = \Phi\left(\frac{\ln x - \lambda}{\zeta}\right)$$

Standard deviation:
$$\zeta = \sqrt{\ln\left(\frac{\sigma^2}{\mu^2} + 1\right)}$$

Mean:
$$\lambda = \ln \mu - \frac{\zeta^2}{2}$$

Matlab - Distribution fitting

Maximum likelihood estimation – Find parameters that maximize the likelihood of the observations

```
namefit(data)
```

Estimates the parameters of the distribution `name` associated with `data`

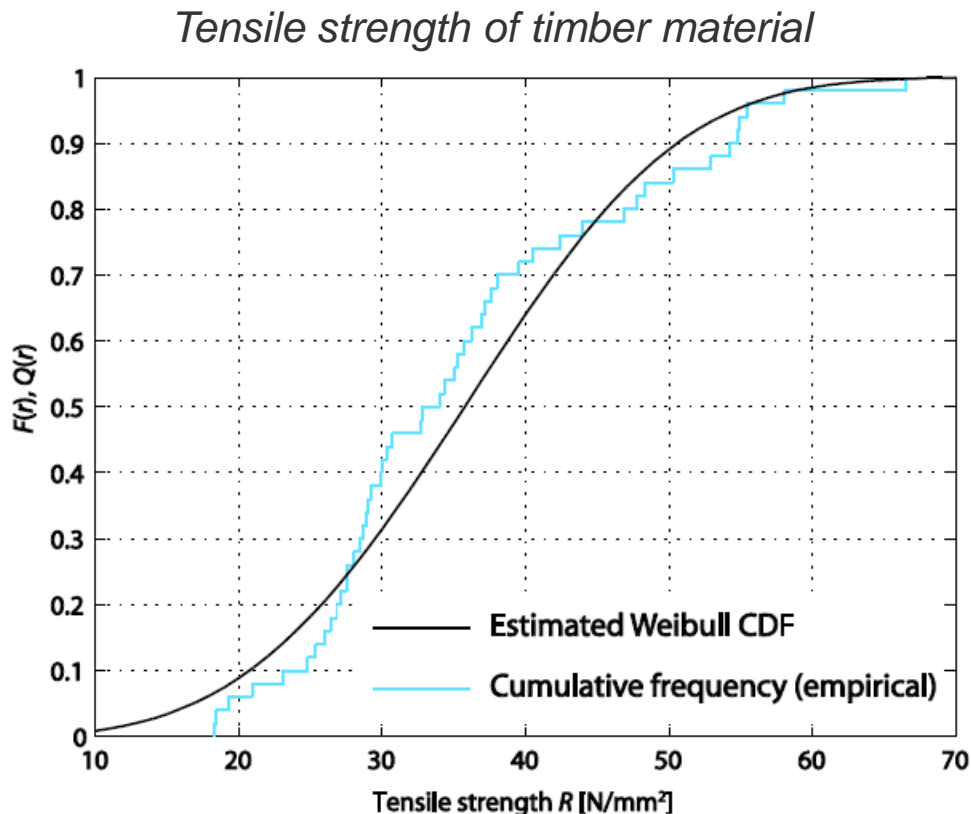Construct a distribution object by fitting to data using `fitdist`

Example:

```
pn = fitdist(data,'norm');
```

Example: wblfit_.m

Graphical methods for distribution fitting

- Plot cumulative frequency diagram (empirical CDF) against CDF of fitted distribution

Tensile strength of timber material

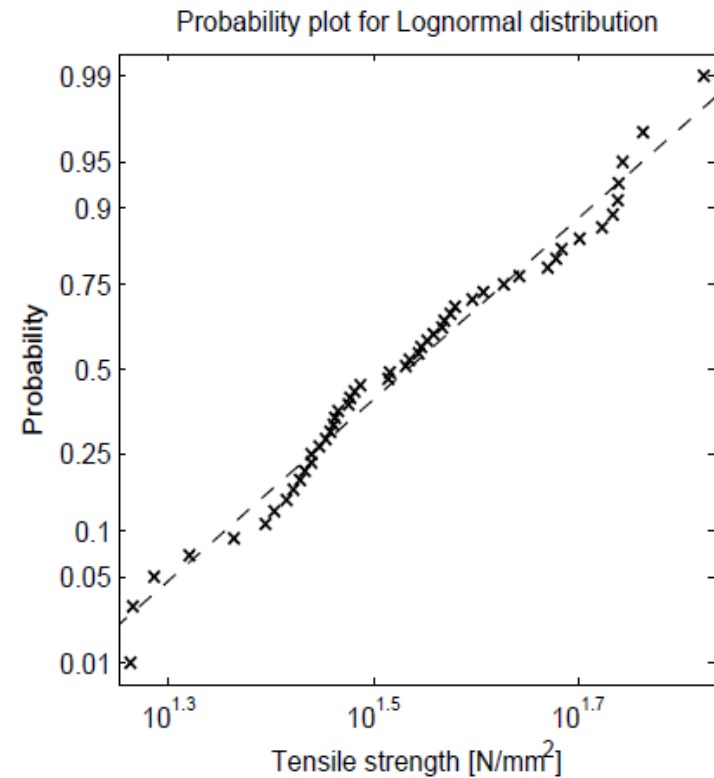# Graphical methods for distribution fitting
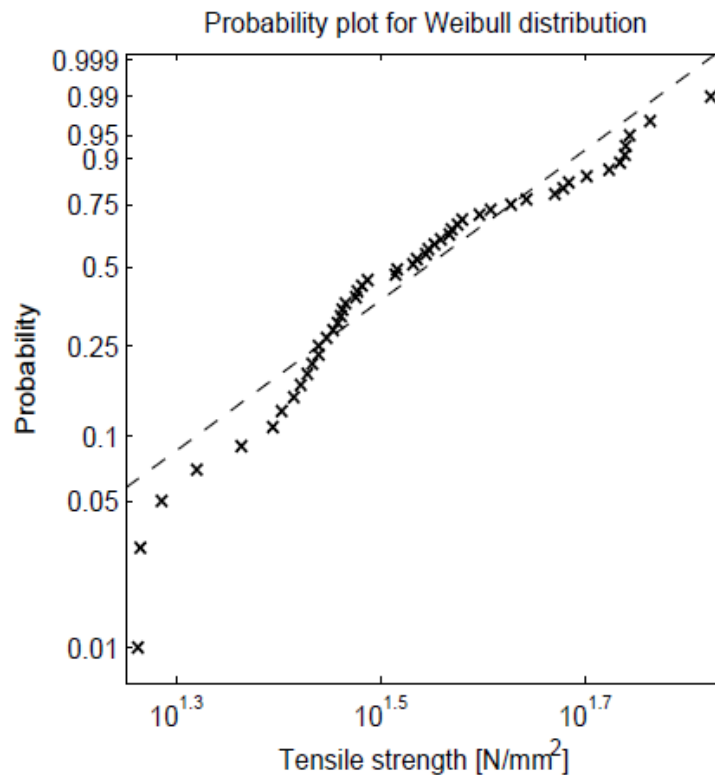
Probability plot

- A quantile plot with scaled axes
- The scaling is selected so that if the data follow exactly a distribution type then the plot is a straight line
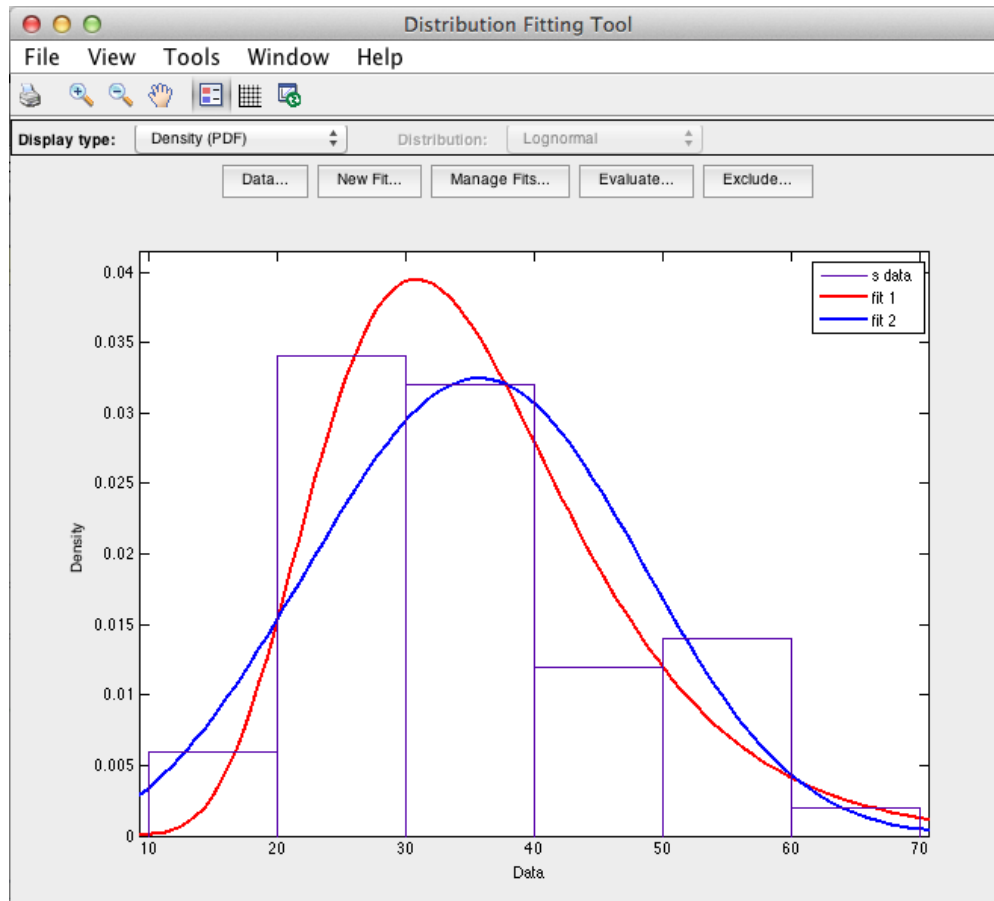
```
probplot('name',data)
```

Graphical methods for distribution fitting

Probability plot

Tensile strength of timber material

Distribution fitting tool

Engineering Data Analysis with Matlab

Group organization for project teams

Please form groups of 2 to 3 students for your project work

$\Rightarrow$ Send an e-mail to anke.scherb@tum.de or tell the tutors during the tutorial
   until May 9

Project work

- Start May 16

- Deadline June 27

- Submit your team's report and Matlab codes

Engineering Data Analysis with Matlab

Project report:

- Structure your report, e.g.: introduction, approach/methods, results, conclusions

- Justify and reinforce your answers and thoughts

- Layout the report and stick to your layout chosen

- Use tables and figures to present your results

- Label all tables and figures in the report

- Always give units and label axis

- Cite correctly and consistently when using literature / internet sources